Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application. The amendments find full support in the originally filed application.

Listing of Claims

1. **(Currently Amended)** A method for planning and scheduling tasks within ~~at least one request~~ a plurality of requests for change ~~(RFC)~~ within a change window in a computing system comprising the steps of:

deciding whether or not ~~an~~ a request for change ~~RFC~~ should be done with respect to maximizing a profit value expressed as a value of performing the plurality of requests for change minus a value of associated costs, wherein ~~the~~ a request for change ~~RFC~~ comprises a set of tasks interrelated by temporal and location-specific dependencies;

for each request for change ~~RFC~~ to be done, assigning individual tasks within each request for change ~~RFC~~ to acceptable servers;

for each request for change ~~RFC~~ to be done, assigning a start time to said individual tasks;

wherein the set of tasks comprises hardware changes and/or software changes;

wherein the change window describes a period of time during which ~~the~~ a request for change ~~RFC~~ is to be done; and

wherein precedence constraints among tasks within ~~the~~ a request for change ~~RFC~~ are enforced.

2. **(Currently Amended)** The method of Claim 1, further comprising the step of reserving all ~~the~~ servers involved for a duration that begins at the start of a first task and ends at the finish of a last task for each request for change ~~RFC~~ that should be done.

3. **(Cancelled)**

4. **(Currently Amended)** The method of Claim 1 further comprising the step of maximizing the number of requests for change ~~RFCs~~ done.

5. **(Original)** The method of Claim 1 further comprising the step of minimizing total downtime.

6. **(Previously Presented)** The method of Claim 1 further comprising the step of minimizing at least one cost associated with downtime.

7. **(Previously Presented)** The method of Claim 1 further comprising the step of minimizing a total execution time in implementing a task.

8. **(Currently Amended)** The method of Claim 1 further comprising the step of maximizing the number of requests for change ~~RFCs~~ meeting their deadlines.

9. **(Currently Amended)** The method of Claim 1 further comprising the step of minimizing multiple deadline penalties associated with requests for change ~~RFCs~~ and/or their respective tasks.

10. **(Currently Amended)** The method of Claim 1 further comprising the step of minimizing an average response time of each request for change ~~RFC~~.

11. **(Currently Amended)** The method of Claim 1 further comprising the step of minimizing a weighted average response time of each request for change ~~RFC~~.

12. **(Currently Amended)** A system for planning and scheduling tasks within ~~at least one request~~ a plurality of requests for change ~~(RFC)~~ within a change window in a computing system, comprising:

a processor;

an arrangement for deciding whether or not ~~an~~ a request for change ~~RFC~~ should be done with respect to maximizing a profit value expressed as a value of performing the plurality of requests for change minus the associated costs, wherein ~~the RFC~~ a request for change comprises a set of tasks interrelated by temporal and location-specific dependencies;

an arrangement for assigning individual tasks to acceptable servers for each request for change ~~RFC~~ to be done; and

an arrangement for assigning a start time to said individual tasks for each request for change ~~RFC~~ to be done;

wherein the set of tasks comprises hardware changes and/or software changes;

wherein the change window describes a period of time during which ~~the~~ a request for change ~~RFC~~ is to be done; and

wherein precedence constraints among tasks within ~~the~~ a request for change ~~RFC~~ are enforced.

13. **(Currently Amended)** The system of Claim 12, further comprising an arrangement for reserving all the servers involved for a duration that begins at the start of the first task and ends at the finish of the last task for each request for change ~~RFC~~ that should be done.

14. **(Cancelled)**

15. **(Currently Amended)** The system of Claim 12, further comprising an arrangement for maximizing the number of requests for change ~~RFCs~~ done.

16. **(Original)** The system of Claim 12, further comprising an arrangement for minimizing total downtime.

17. **(Previously Presented)** The system of Claim 12, further comprising an arrangement for minimizing at least one cost associated with downtime.

18. **(Previously Presented)** The system of Claim 12, further comprising an arrangement for minimizing a total execution time in implementing a task.

19. **(Currently Amended)** The system of Claim 12, further comprising an arrangement for maximizing the number of requests for change ~~RFCs~~ meeting their deadlines

20. **(Currently Amended)** The system of Claim 12, further comprising an arrangement for minimizing multiple deadline penalties associated with requests for change ~~RFCs~~ and/or their respective tasks.

21. **(Currently Amended)** The system of Claim 12, further comprising an arrangement for minimizing an average response time of each request for change ~~RFC~~.

22. **(Currently Amended)** The system of Claim 12, further comprising an arrangement for minimizing a weighted average response time of each request for change ~~RFC~~.

23. **(Currently Amended)** A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform a method for planning and scheduling tasks within ~~at least one request~~ a plurality of requests for change ~~(RFC)~~ within a change window in a computing system, the method comprising the steps of:

deciding whether or not ~~an~~ a request for change ~~RFC~~ should be done with respect to maximizing a profit value expressed as a value of performing the plurality of requests for change minus a value of associated costs, wherein ~~the~~ a request for change ~~RFC~~ comprises a set of tasks interrelated by temporal and location-specific dependencies;

for each request for change ~~RFC~~ to be done, assigning individual tasks within each request for change ~~RFC~~ to acceptable servers;

for each request for change ~~RFC~~ to be done, assigning a start time to said individual tasks;

wherein the set of tasks comprises hardware changes and/or software changes;

wherein the change window describes a period of time during which ~~the~~ a request for change ~~RFC~~ is to be done; and

wherein precedence constraints among tasks within ~~the~~ a request for change ~~RFC~~ are enforced.

24. **(New)** The program storage device according to claim 23, wherein:

start-to-finish constraints between tasks of a request for change are enforced;

finish-to-finish constraints between tasks of a request for change are enforced;

colocation task/server assignment constraints are enforced for all requests for change;

exlocation task/server assignment constraints are enforced for all requests for change;

resource capacity constraints are enforced on each acceptable server executing one or more tasks for one or more requests for change;

requests for change with a deadline that falls within the change window must be performed;

each task of a request for change that is performed is assigned to a single acceptable server;

no acceptable server can work on more than one task at any time; and

all tasks of all requests for change that are performed must be performed during the change window.

25. **(New)** The method according to claim 1, wherein:

start-to-finish constraints between tasks of a request for change are enforced;

finish-to-finish constraints between tasks of a request for change are enforced;

colocation task/server assignment constraints are enforced for all requests for change;

exlocation task/server assignment constraints are enforced for all requests for change;

resource capacity constraints are enforced on each acceptable server executing one or more tasks for one or more requests for change;

requests for change with a deadline that falls within the change window must be performed;

each task of a request for change that is performed is assigned to a single acceptable server;

no acceptable server can work on more than one task at any time; and

all tasks of all requests for change that are performed must be performed during the change window.